SearchSploit Security Assessment Project

Complete Guide for Vulnerability Research and Exploit Database Analysis

Table of Contents

Project Overview

What This Project Covers

This comprehensive guide demonstrates the use of SearchSploit for vulnerability research, exploit database analysis, and security assessment within a controlled Kali Linux environment running on Oracle VirtualBox. The project follows industry standards and compliance frameworks to ensure professional-grade security analysis.

Learning Objectives

- Master SearchSploit for vulnerability research
- Understand exploit database navigation and analysis
- Identify security vulnerabilities in software and services
- Apply security frameworks (NIST CSF, ISO 27001)

- Develop professional security assessment skills
- Learn to correlate vulnerability information with real-world threats

Definition and Purpose

What is SearchSploit?

SearchSploit is a command-line tool that serves as an offline interface to the Exploit Database (Exploit-DB). It allows security professionals and penetration testers to search for exploits, shellcodes, and vulnerability information without requiring an internet connection.

Core Components

1. **Exploit Database**: A comprehensive archive of public exploits and corresponding vulnerable software
2. **Command-Line Interface**: Efficient search and filtering capabilities
3. **Local Repository**: Offline access to thousands of exploits
4. **Integration Capabilities**: Works seamlessly with other security tools

Primary Use Cases

- **Vulnerability Research**: Identifying known vulnerabilities in target systems
- **Penetration Testing**: Finding applicable exploits during authorized assessments
- **Security Auditing**: Verifying patch levels and security posture
- **Threat Intelligence**: Understanding attack vectors and exploit availability
- **Security Training**: Learning about vulnerability exploitation techniques

Importance in Cybersecurity

SearchSploit is critical because it:

- Provides immediate access to exploit information without internet dependency
- Helps security professionals stay informed about vulnerabilities
- Enables proactive security measures by identifying potential attack vectors
- Supports compliance requirements for vulnerability management

- Facilitates faster incident response and threat assessment

Prerequisites and Setup

System Requirements

Hardware Requirements

- **RAM**: Minimum 2GB (4GB recommended)
- **Storage**: 20GB available space
- **Processor**: 64-bit dual-core or better

Software Requirements

- **Operating System**: Kali Linux (latest version recommended)
- **Virtualization**: Oracle VirtualBox 6.0 or higher
- **Network**: NAT or Bridged network configuration

Installation Steps

Step 1: Verify Kali Linux Installation

bash

Check Kali version

cat /etc/os-release

Update system

sudo apt update && sudo apt upgrade -y

Step 2: Verify SearchSploit Installation

SearchSploit comes pre-installed with Kali Linux. Verify installation:

bash

Check SearchSploit version

searchsploit --version

Check installation path

which searchsploit

View help menu

searchsploit --help

```
┌──(bjnetwork㉿bjnetwork)-[~]
└─$ searchsploit --version
/usr/bin/searchsploit: illegal option -- -
  Usage: searchsploit [options] term1 [term2] ... [termN]

==========
 Examples
==========
  searchsploit afd windows local
  searchsploit -t oracle windows
  searchsploit -p 39446
  searchsploit linux kernel 3.2 --exclude="(PoC)|/dos/"
  searchsploit -s Apache Struts 2.0.0
  searchsploit linux reverse password
  searchsploit -j 55555 | jq
  searchsploit --cve 2021-44228

  For more examples, see the manual: https://www.exploit-db.com/searchsploit

==========
 Options
==========
## Search Terms
   -c, --case     [term]      Perform a case-sensitive search (Default is inSE
   -e, --exact    [term]      Perform an EXACT & order match on exploit title
                                e.g. "WordPress 4.1" would not be detect "Word
   -s, --strict               Perform a strict search, so input values must ex
                                e.g. "1.1" would not be detected in "1.0 < 1.3
   -t, --title    [term]      Search JUST the exploit title (Default is title
       --exclude="term"       Remove values from results. By using "|" to sepa
                                e.g. --exclude="term1|term2|term3"
       --cve      [CVE]       Search for Common Vulnerabilities and Exposures
```

Step 3: Update Exploit Database

bash

Update the local exploit database

sudo searchsploit -u

Verify update

ls -lh /usr/share/exploitdb/

Step 4: Configure Working Environment

bash

Create project directory

mkdir -p ~/searchsploit-project/{reports,evidence,logs}

Navigate to project directory

cd ~/searchsploit-project

Create documentation file

touch assessment-notes.txt

Step-by-Step Implementation

Phase 1: Environment Preparation

Step 1.1: System Verification

bash

Verify network connectivity

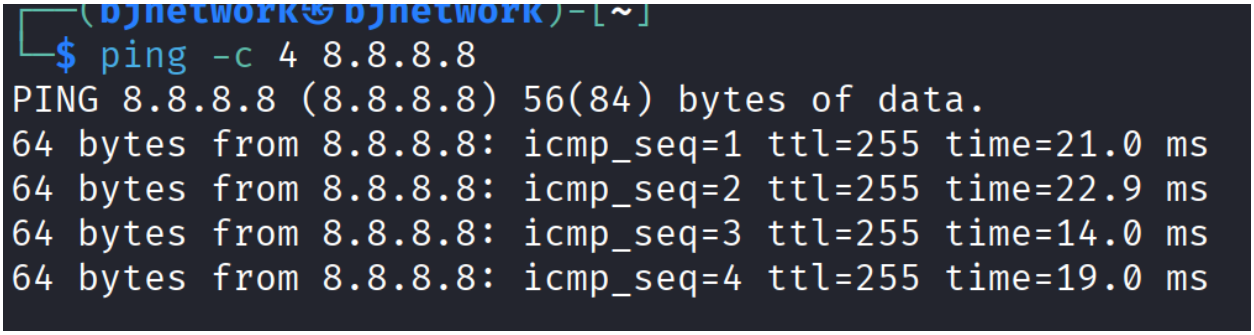
ping -c 4 8.8.8.8

Check available disk space

df -h

Verify exploit database location

ls -lah /usr/share/exploitdb/exploits/



Step 1.2: Database Statistics

bash

Count total exploits

searchsploit --stats

View database path

searchsploit --path

Phase 2: Basic SearchSploit Operations

Step 2.1: Simple Search Operations

bash

Search for specific software

searchsploit apache

```
└─$ searchsploit apache
```

Exploit Title	Path
Apache (Windows x86) - Chunked Encoding (Metasploit)	windows_x86/remote/16782.rb
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution	php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner	php/remote/29316.py
Apache - Arbitrary Long HTTP Headers (Denial of Service)	multiple/dos/360.pl
Apache - Arbitrary Long HTTP Headers Denial of Service	linux/dos/371.c

Search for specific version

searchsploit apache 2.4

```
└─$ searchsploit apache 2.4
```

Exploit Title	Path
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution	php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner	php/remote/29316.py
Apache 2.2.4 - 413 Error HTTP Request Method Cross-Site Scripting	unix/remote/30835.sh
Apache 2.4.17 - Denial of Service	windows/dos/39037.php
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local Privilege Escalation	linux/local/46676.php
Apache 2.4.23 mod_http2 - Denial of Service	linux/dos/40909.py

Search for specific platform

searchsploit windows

```
┌──(bjnetwork㉿bjnetwork)-[~]
└─$ searchsploit windows
```

Exploit Title	Path
(Gabriel's FTP Server) Open & Compact FTP Server 1.2 - 'PORT' Remote Denial of Service	windows/dos/12698.py
(Gabriel's FTP Server) Open & Compact FTP Server 1.2 - Authentication Bypass / Directory Traversal SAM Re	windows/remote/27401.py
(Gabriel's FTP Server) Open & Compact FTP Server 1.2 - Full System Access	windows/remote/13932.py
(Gabriel's FTP Server) Open & Compact FTP Server 1.2 - Universal Denial of Service	windows/dos/12741.py

Case-insensitive search

searchsploit -t apache

Step 2.2: Advanced Search Techniques

bash

Search with multiple terms

searchsploit apache remote

```
└─$ searchsploit apache remote
```

Exploit Title	Path
ActiveMQ < 5.14.0 - Web Shell Upload (Metasploit)	java/remote/42283.rb
Apache (Windows x86) - Chunked Encoding (Metasploit)	windows_x86/remote/16782.rb
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution	php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner	php/remote/29316.py
Apache - httpOnly Cookie Disclosure	multiple/remote/18442.html

Exclude terms from search

searchsploit apache --exclude="2.2"

Search in exploit title only

searchsploit -t "privilege escalation"

Search for exact match

searchsploit --exact "Apache 2.4.49"

Step 2.3: Filtering Results

bash

Filter by platform

searchsploit openssh --platform=linux

Filter by type

searchsploit mysql --type=webapps

Combine filters

searchsploit windows --platform=windows --type=local

Phase 3: Examining Exploits

Step 3.1: Viewing Exploit Details

bash

Examine specific exploit

searchsploit -x exploits/linux/remote/12345.txt

Copy exploit to current directory

searchsploit -m exploits/linux/remote/12345.txt

Mirror multiple exploits

searchsploit -m 12345 12346 12347

Step 3.2: Exploit Analysis

bash

View exploit metadata

head -n 50 12345.txt

Search for CVE information

grep -i "CVE" 12345.txt

Check exploit requirements

grep -i "requirement\|prerequisite" 12345.txt

Phase 4: Integration with Nmap Results

Step 4.1: Service Scanning

bash

Perform service scan on target

nmap -sV -oX scan_results.xml 192.168.1.100

Save scan results

nmap -sV -oN scan_output.txt 192.168.1.100

Step 4.2: Automated Vulnerability Matching

bash

Use Nmap XML output with SearchSploit

searchsploit --nmap scan_results.xml

Generate detailed report

searchsploit --nmap scan_results.xml --colour > vulnerability_report.txt

Phase 5: CVE Cross-Referencing

Step 5.1: CVE Database Search

bash

Search by CVE number

searchsploit --cve CVE-2021-44228

Search multiple CVEs

searchsploit --cve CVE-2021-3156

Export CVE information

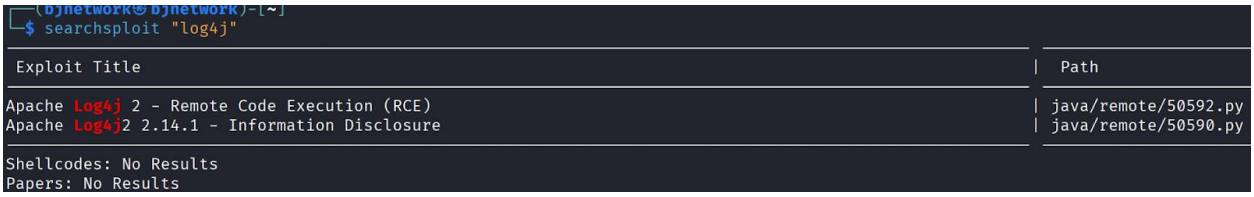
searchsploit --cve CVE-2021-44228 --json > cve_results.json

Step 5.2: Vulnerability Correlation

bash

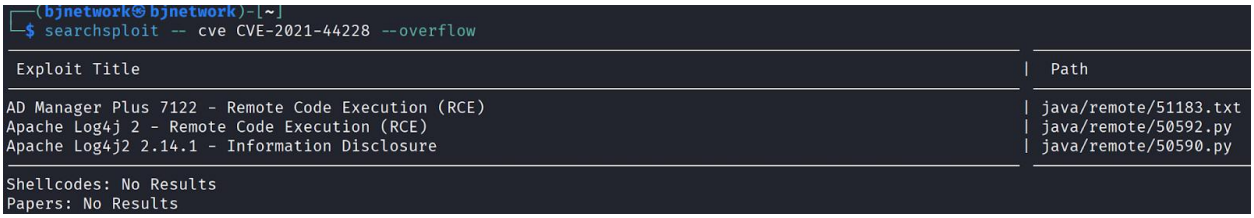
Search for vulnerability name

searchsploit "log4j"



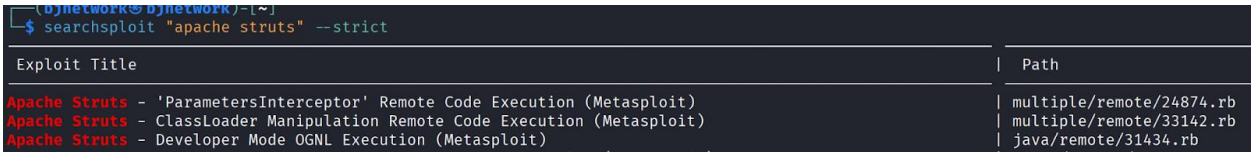
Cross-reference with CVE

searchsploit --cve CVE-2021-44228 --overflow



Check for related exploits

searchsploit "apache struts" --strict



Phase 6: Report Generation

Step 6.1: JSON Output

bash

Generate JSON report

searchsploit apache --json > apache_exploits.json

Pretty print JSON

cat apache_exploits.json | python3 -m json.tool

Filter JSON output

searchsploit windows privilege --json | jq '.RESULTS_EXPLOIT[]'

Step 6.2: XML Output

bash

Generate XML report

searchsploit mysql --xml > mysql_exploits.xml

View XML structure

xmllint --format mysql_exploits.xml | head -n 50

Step 6.3: Comprehensive Documentation

bash

Create detailed assessment report

searchsploit openssh --colour --overflow > openssh_assessment.txt

Add timestamp to report

echo "Assessment Date: $(date)" >> openssh_assessment.txt

Append system information

echo "Target: 192.168.1.100" >> openssh_assessment.txt

Commands Reference

Essential SearchSploit Commands

Basic Search Commands

Command	Purpose	Example
searchsploit <term>	Basic search	searchsploit apache
searchsploit -t <term>	Search title only	searchsploit -t "sql injection"
searchsploit --strict	Strict search mode	searchsploit --strict apache 2.4
searchsploit --exact	Exact match search	searchsploit --exact "Apache 2.4.49"

Filter Commands

Command	Purpose	Example
--exclude="term"	Exclude results	searchsploit apache --exclude="2.2"
--platform=<os>	Filter by platform	searchsploit --platform=linux
--type=<type>	Filter by type	searchsploit --type=remote
--overflow	Show long titles	searchsploit --overflow

Output Commands

Command	Purpose	Example
-x <id>	Examine exploit	searchsploit -x 12345
-m <id>	Mirror/copy exploit	searchsploit -m 12345
--json	JSON output	searchsploit apache --json

--xml	XML output	searchsploit mysql --xml
--www	Online URL	searchsploit apache --www

Integration Commands

Command	Purpose	Example
--nmap <file>	Parse Nmap XML	searchsploit --nmap scan.xml
--cve <CVE>	Search by CVE	searchsploit --cve CVE-2021-44228
-u	Update database	searchsploit -u
--path	Show database path	searchsploit --path

Advanced Search Patterns

bash

Multiple term search (AND logic)

searchsploit term1 term2

Search with wildcards

searchsploit apache*

Search specific versions

searchsploit "apache 2.4.[0-9]"

Combine multiple filters

searchsploit windows privilege --platform=windows --type=local --exclude="vista"

Practical Command Workflows

Workflow 1: Service Assessment

bash

Step 1: Scan target

nmap -sV -oX target_scan.xml 192.168.1.100

Step 2: Cross-reference with exploits

searchsploit --nmap target_scan.xml

Step 3: Export findings

searchsploit --nmap target_scan.xml --json > findings.json

Step 4: Review critical vulnerabilities

searchsploit --nmap target_scan.xml | grep -i "remote\|critical"

Workflow 2: CVE Research

bash

Step 1: Search CVE

searchsploit --cve CVE-2021-44228

Step 2: Examine exploit details

searchsploit --cve CVE-2021-44228 -x

Step 3: Copy exploit locally

searchsploit --cve CVE-2021-44228 -m

Step 4: Document findings

searchsploit --cve CVE-2021-44228 --overflow > cve_analysis.txt

Workflow 3: Software Audit

bash

Step 1: Identify software version

searchsploit "Apache 2.4.49"

Step 2: Review all versions

searchsploit apache 2.4

Step 3: Filter for critical exploits

searchsploit apache 2.4 --type=remote

Step 4: Generate comprehensive report

searchsploit apache 2.4 --json | jq '.RESULTS_EXPLOIT[] | select(.Type == "remote")'